

N89 - 10098

DYNAMIC TASK ALLOCATION FOR A MAN-MACHINE SYMBIOTIC SYSTEM*

L. E. Parker and F. G. Pin

Center for Engineering Systems Advanced Research
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831-6364

"The submitted manuscript has been authored by a contractor of the U.S. Government under contract No. DE-AC05-84OR21400. Accordingly, the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes."

Paper Submitted to:

Goddard Conference on Space Applications
of Artificial Intelligence and Robotics
May 13-14, 1987
Greenbelt, Maryland

*Research sponsored by the Engineering Research Program of the Office of Basic Energy Sciences, of the U.S. Department of Energy, under contract No. DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc.

ORIGINAL PAGE IS
OF POOR QUALITY

ABSTRACT

This paper presents a methodological approach to the dynamic allocation of tasks in a man-machine symbiotic system in the context of dexterous manipulation and teleoperation. This paper addresses symbiosis containing two symbiotic partners which work toward controlling a single manipulator arm for the execution of a series of sequential manipulation tasks. The proposed automated task allocator uses knowledge about the constraints/criteria of the problem, the available resources, the tasks to be performed, and the environment to dynamically allocate tasks to the man and the machine. The presentation of the methodology includes discussions concerning the characteristics of the man-machine symbiotic system, the interaction of the knowledge areas, the flow of execution, and the dynamic nature of the task allocation.

1.0. INTRODUCTION

During the last few decades, there has been a growing awareness and belief that automation-related technologies and intelligent machines will play an increasing role in improving the development and operation of complex and advanced systems. In this context, research and development has taken place on a broad range of technologies aimed at achieving automated systems varying from fully remotely-controlled systems such as advanced teleoperators and servomanipulators to fully autonomous intelligent robots involving artificial intelligence, super-computing, machine vision, and advanced control. Within this large spectrum of technological research, work has recently been initiated on what is proposed to be a new class of automated systems which appear promising for improving the productivity, quality, and safety of operation of advanced systems. This new type of automated system is referred to as "Man-Machine Symbiosis" and would utilize the concepts of machine intelligence and remote-control technology to achieve full man-machine cooperative control and intelligence [2].

The ultimate function of such symbiotic systems would be to dynamically optimize the division of work between the man and the machine and to facilitate their cooperation through shared knowledge, skills, and experiences. The optimization of the man-machine partnership in both the electromotive and intellectual domain would be realized by coupling a dynamic allocation of tasks between the human and the machine with an embedded system learning capability to allow the machine, an intelligent robotic system, to learn new tasks through assimilation of experience and observation of the human [3], [4], [5].

This paper presents a methodological approach to the dynamic allocation of tasks for a man-machine symbiotic system in a simplified case of dexterous manipulation and teleoperation. In this formulation, two symbiotic partners are considered: a human teleoperator and an intelligent robotic system. Both partners work toward controlling a single manipulator arm for the execution of a series of sequential manipulation tasks. Section 2 of the paper outlines the characteristics of the specific man-robot symbiont considered here, while section 3 presents a generalized task allocation procedure. For an example illustrating the results of the conceptual architecture in the context of remote manipulation, refer to [7].

2.0 CHARACTERISTICS OF A MAN-MACHINE SYMBIOTIC SYSTEM

The man-machine system addressed in this paper consists of two symbiotic partners, a human teleoperator and an intelligent robot system with its controller, which cooperate to perform a series of sequential manipulation tasks involving a single manipulator arm. To facilitate the division of work between the man and the robot, several automated modules are proposed to be incorporated into the system to perform responsibilities such as task subdivision, analysis, and allocation. Such a scenario can be depicted as shown in figure 1.

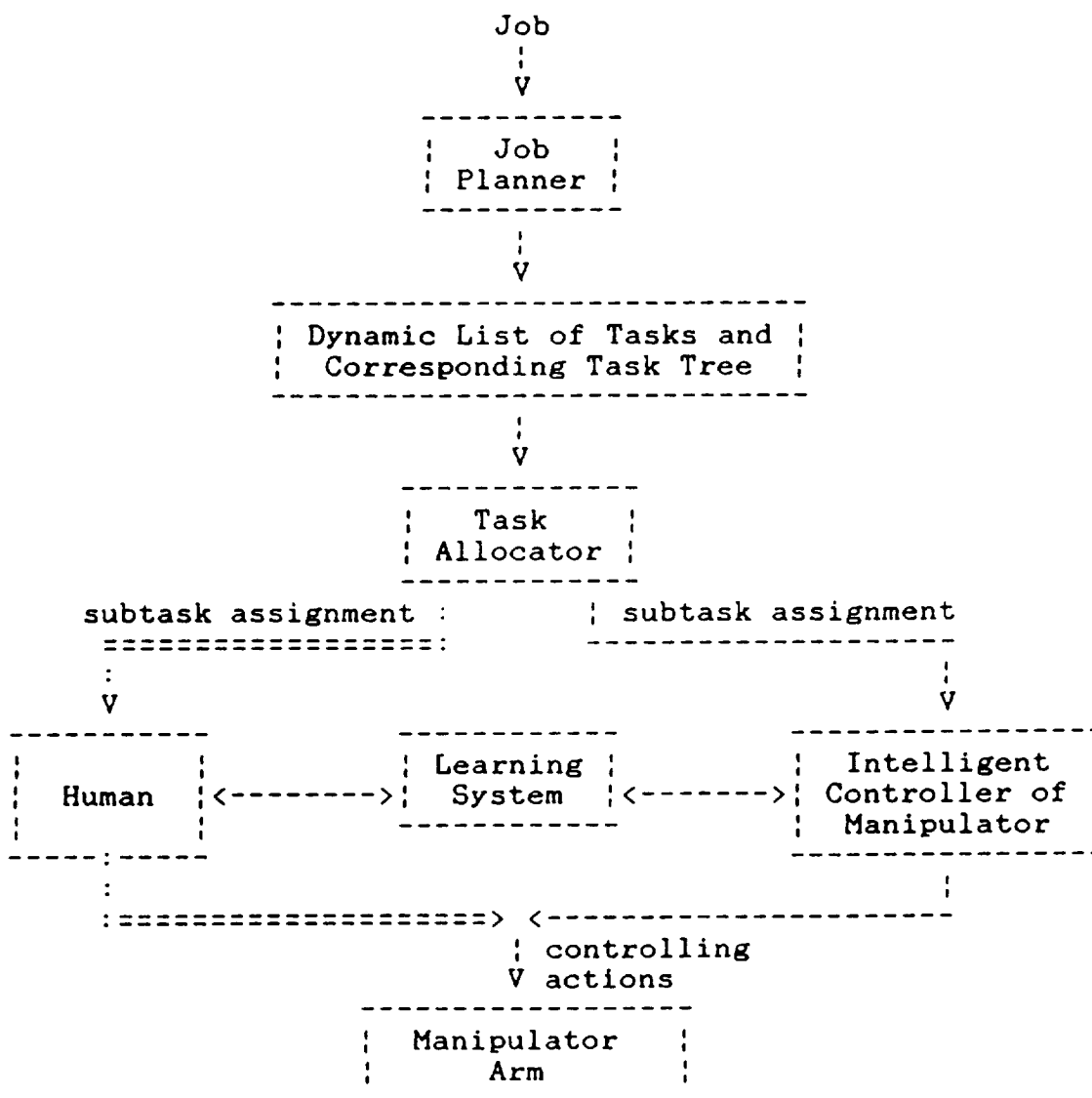


Figure 1

A job planner is responsible for decomposing the overall job to be performed (such as INSTALL ELECTRICAL EQUIPMENT) into its component lower-level subtasks (such as FIND WRENCH or GRASP WRENCH), indicating the order in which the subtasks must be performed. The resulting task decomposition tree (see section 3.1.3), is passed to the task allocator, which assigns a subtask either to the human or to the intelligent robot controller of the manipulator. The human or the intelligent robot controller then sends controlling actions to the manipulator arm for execution of the subtask. To improve its performance and to increase its range of capabilities, the intelligent robot controller of the manipulator arm must ultimately use an embedded learning system to learn new tasks through assimilation of experience, observation of the human, and direct instruction [3], [4], [5].

This paper is concerned only with the task allocator and its relationship to the other entities in the man-machine symbiotic system. This paper assumes that a complete description of the tasks to be performed is provided to the task allocator by either the human or an automated system. Research is currently being performed on automating the job planner. This paper also does not discuss any details related to the embedded learning system, which is currently being researched and will be discussed in future publications.

To determine the necessary characteristics of the task allocator in this symbiotic system, one can first observe that both intelligent resources (the human and the intelligent controller of the manipulator arm) are using the same medium (the manipulator arm) to execute the subtasks. The manipulator arm actuator can receive and respond to commands from a single source at any instant in time. Consequently, the human and the intelligent robot controller cannot command the arm simultaneously or independently. Therefore, the task allocator must deal with the allocation of sequential manipulation tasks, rather than concurrent tasks. However, it is likely that while the human or the machine is performing a subtask with the manipulator arm, other actions are occurring in the background, such as monitoring of the task execution, world modeling, planning, and learning. This aspect is necessary in order for the symbiotic system to function effectively. Nevertheless, as a first step, this work will focus on the sequential task problem of allocating a series of sequential manipulation subtasks to the man and the machine. Research is currently underway to extend this methodology to allow the human

and/or the machine to perform additional subtasks which compete for their time while the manipulation subtasks are being performed.

Another essential requirement of the task allocator in this man-machine system is its ability to be event-driven, responding to changes in the work constraints, physical environment, or unexpected events by altering the task allocation to adjust to new conditions. This dynamic nature of the task allocator allows the man-machine symbiont to cope with a changing environment, causing the resource most appropriate for performing a subtask to be assigned the subtask. In order for a dynamic allocation of subtasks to be successful, the human and the intelligent controller of the manipulator arm must be able to perform at least some of the subtasks interchangeably; otherwise, the allocation can be automatically pre-determined simply by assigning each subtask to the only resource that is able to perform it. Such a static allocation of subtasks is intolerant of faults, for if one resource failed in performing its subtask, another resource could not take over the operation of that subtask. The dynamic allocation of subtasks, however, does not usually suffer from this symptom, and can result in an effective use of the resources which is more tolerant to resource faults [1]. Note that even the dynamic method of task allocation will not be completely intolerant to resource faults during the execution of subtasks which can only be performed by one specific resource.

In summary, the task allocator in this symbiotic system must deal with the dynamic allocation of sequential manipulation subtasks to two resources, a human and an intelligent robot controller, responding to events during the subtask execution which lead to a reallocation of subtasks. The remainder of this paper will address a task allocation methodology having these characteristics.

3.0 DYNAMIC TASK ALLOCATION METHODOLOGY

3.1 KNOWLEDGE AREAS

The purpose of the task allocator in man-machine symbiosis is to attempt to dynamically optimize the division of work between the man and the machine. Since the exact interpretation of "optimal division of work" must be allowed to vary according to the requirements of each individual problem scenario, the task allocator must know what constraints and criteria are placed on the task allocation, what the requirements of the subtasks are, and information concerning the characteristics of the environment in which the problem is to be solved. The task allocator must also

have information about the capabilities of the human and the intelligent robot controller to determine the resource which is most appropriate for performing a subtask in a given scenario. The knowledge about these areas can be categorized into four main knowledge bases which are described in the following sections.

3.1.1 CONSTRAINTS/CRITERIA

The constraints/criteria are determined by a source external to the task allocator and place performance measures, limitations, restrictions, and/or regulations on the task allocation problem solution. The intent of the constraints/criteria is to alter the task allocation strategy to adapt to differing problem contexts. The task allocator must adhere to these constraints/criteria in determining the task allocation. These limitations may prevent the use of certain resources for some subtasks, or may mandate the use of certain resources for other subtasks.

Examples of possible constraints/criteria are as follows:

- minimize time of job completion
- maximize quality of result
- minimize human involvement (e.g. in a hazardous environment or to prevent boredom or fatigue)

The task allocator must know how to handle any constraint that is placed on the solution. For example, if the constraint is to minimize the time of task completion, the task allocator must compute the estimated time each resource will take to complete a subtask (refer to sections 3.1.2 and 3.1.3 for further information) and then assign the subtask to the resource requiring the lesser time. For each application of the task allocator, certain constraints/criteria are initially in effect while other constraints/criteria are ignored. Although this paper only deals with situations having one constraint in effect at a time, this methodology has the potential for being extended to handle combinations of several constraints/criteria for the optimization of the solution.

3.1.2 RESOURCES

In this paper, resources are defined to be intelligent entities (such as humans or computers) which are available for performing subtasks to solve a problem, or to achieve a goal. In this paper, only two resources are considered: a human and an intelligent robot controller. Obviously, the task allocator must have some information concerning the available resources before it can begin the job of task allocation. The task allocator must know what capabilities

each of the resources possess, how well the resources use their capabilities in performing subtasks, how timely the resources use their capabilities to perform the subtasks, and the current status of the resources (i.e., when each resource will be available to perform subtasks). The capabilities of the resources are defined in this paper to be either the abilities the resources have to perform certain physical actions, or the knowledge the resources have of certain objects. The capabilities can be defined as needed for particular applications, and could include physical abilities such as MANIPULATION or VISION, or knowledge of objects, such as WRENCH or BOLT.

Each resource can have many capabilities. However, a resource will probably not have the same level of achievement of each of its capabilities, and it certainly will not exercise each capability with identical speeds. For example, although a human has capabilities of both COMPUTATION and VISION, he probably can examine a photograph (using VISION) much easier and better than he can add a few numbers in his head (using COMPUTATION). On the other hand, a computer may also have capabilities of COMPUTATION and VISION, yet it is much more difficult for it to examine a photograph than it is for it to add a few numbers.

The knowledge about the capabilities of the resources is initially given to the task allocator as input. The actual information stored about the capabilities of the resources is directly related to the constraints which might at some time be present in the problem scenario. For example, the constraint "minimize time of task completion" requires that "timeliness of achievement" factors be provided, while the constraint "maximize quality of result" requires that "level of achievement" factors be provided. Additional constraints placed on the problem may require the storage of further information on the capabilities of the resources.

Although the knowledge about the capabilities is quantified differently depending upon whether the capability refers to a physical ability or to a knowledge about an object, one evaluation number is obtained for each factor (such as level of achievement and timeliness of achievement) of each capability. The evaluation numbers are then used to help determine the appropriate task allocation. If the capability refers to a physical ability, the evaluation number indicates the skill with which the ability is performed, perhaps on a scale from 0 to 10, or from "unacceptable" to "superior". If the capability refers to a knowledge about an object, the evaluation number indicates how complete the knowledge of that object is, perhaps on a scale from 0 to 10, or from "unknown" to "always known". Depending on the constraints of the given problem and the

subtasks to be performed, the task allocator can select the suitable resources to perform the subtasks based on the characteristics of the resources. This is done by determining what capabilities are required to complete each subtask, finding the available resources which possess the required capabilities, and applying the constraints/criteria of the problem to compute the optimal allocation.

The task allocator would thus have information as follows for the resources:

| <u>resource</u> | <u>capa-</u> <u>bility</u> | <u>level of</u> <u>achievement</u> | <u>timeliness of</u> <u>achievement</u> | <u>availability</u> |
|-----------------|-------------------------------|---------------------------------------|--|---------------------|
| R1 | a11 | l11 | t11 | w units |
| | a12 | l12 | t12 | x units |
| | . | . | . | . |
| | . | . | . | . |
| | a1n | l1n | t1n | y units |
| R2 | a21 | l21 | t21 | w units |
| | a22 | l22 | t22 | x units |
| | . | . | . | . |
| | . | . | . | . |
| | a2n | l2n | t2n | y units |
| | | | | |
| Rm | am1 | lm1 | tm1 | w units |
| | am2 | lm2 | tm2 | x units |
| | . | . | . | . |
| | . | . | . | . |
| | amn | lmn | tmn | y units |

For example, information which could be obtained from a table such as this is as follows:

- o The human has the capability of VISION, can perform VISION on a level of 10 (or "superior") with a "timeliness factor" of 2 (or "extremely fast"), and is currently available to perform VISION.
- o The human has the capability of MANIPULATION, can perform MANIPULATION on a level of 7 (or "fairly good") with a timeliness factor of 4 (or "fairly fast"), but is not currently available to perform MANIPULATION. The

human will be available to perform MANIPULATION in 3 time units.

- o The computer has the capability to RECOGNIZE WRENCH, can RECOGNIZE WRENCH on a level of 4 ("sometimes known") with a timeliness factor of 7 ("fairly slow") , and is currently available to RECOGNIZE WRENCH.

Some important observations can be made in examining this table. First, a resource can have more than one capability available at a time, and it can also use more than one capability at a time in the execution of a subtask. The use of more than one capability at a time should not be confused with the execution of more than one subtask at a time. The resource will only be performing one subtask at once, although it may use several capabilities to accomplish that subtask. For instance, a concurrent computer can use one processor for the capability VISION and another processor for the capability COMPUTATION. Likewise, humans can use the capability of VISION while using the capability of MANIPULATION to hammer a nail. Thus, the use of one capability of a resource does not necessarily mean that the other capabilities of that resource are inaccessible.

The second observation from examination of the table is that since only two resources are considered in this paper (a human and a machine), the above table in an actual application would have only two entries: R1 and R2. However, the extension to m resources is possible and would allow many resources to be considered in the execution of the sequential manipulation subtasks.

3.1.3 TASKS

A job planner must analyze and decompose the job to be performed into its component tasks, subtasks, and sub-subtasks. The role of the job planner can be fulfilled by either the human or an automated job planning system. The current paper does not address the operation of the job planner and assumes that the task breakdown is available as input to the task allocator. An automated job planner for the system will be addressed in a companion publication.

A typical task breakdown tree is shown in figure 2a.

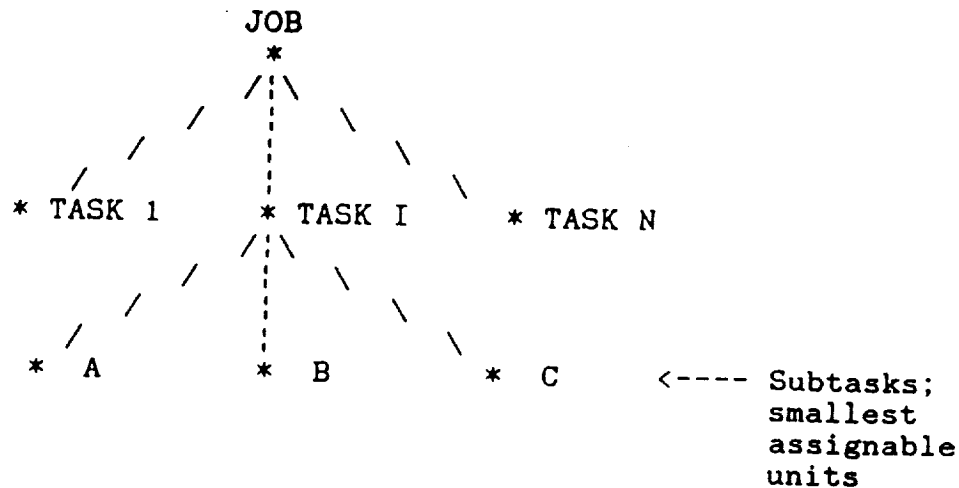


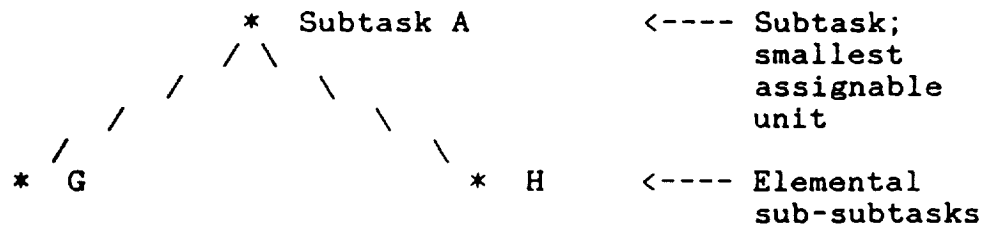
Figure 2a

The job is the highest-level description of a series of related tasks to be performed, such as ASSEMBLE MODULE. The job is decomposed into several tasks, such as INSERT ROD, which must be successfully completed by the resources in order to solve a problem, or to achieve a goal. Each task can be performed entirely by the human, entirely by the computer, or by the human and computer in cooperation. Each task is subdivided as much as needed until the smallest assignable units, or subtasks, are reached. These subtasks are the smallest units that can be feasibly assigned to a single resource. For example, a task UNPLUG CABLE could consist of subtasks FIND CABLE, MOVE TO CABLE, GRASP CABLE, and PULL CABLE. It would be senseless to assign smaller components of these subtasks to more than one resource. The concept of a "smallest assignable unit" is very important since it represents the smallest subdivision of the elements of a task which correlate with the physical mechanics of the actual operation of the symbiotic resources. The definitions of resources, capabilities, and smallest assignable units are, in general, system and task domain dependent.

In order to allocate the subtasks, the task allocator must know what capabilities are required to perform the subtasks and any merit factors associated with each capability. Due to the considerable differences between the intelligent robot controller and the human, the capabilities required for one of these resources to perform a subtask may be very different from those required by the other resource. Because of this, the subtasks must be further subdivided for each resource down to the elemental sub-subtasks which can

be characterized by one or more capabilities and merit factors which are independent of the environment or the context of the problem. An example of the subdivision is shown in figure 2b.

For R1:



For R2:

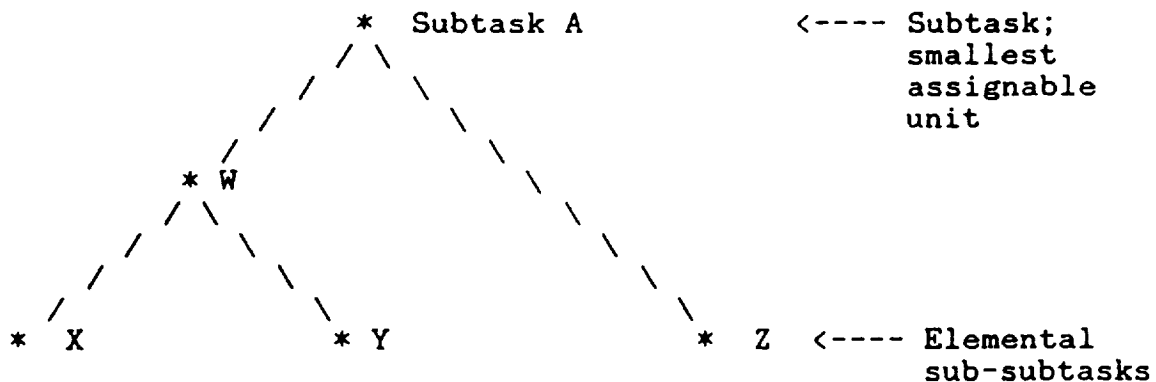


Figure 2b

The list of capabilities required for each subtask is obtained by traversing the lowest-level nodes (leaves), or elemental sub-subtasks, below the subtask in the task breakdown tree, noting all the capabilities required for the lowest-level nodes, or elemental sub-subtasks. This traversal must be performed for each resource, since the resources have different sub-subtask breakdowns, as shown in figure 2b. The merit factor associated with each capability indicates the importance of that capability in the successful performance of the elemental sub-subtask, relative to the other required capabilities. The merit factors are obtained for the capabilities in a manner similar to how the list of required capabilities is obtained -- by traversing the leaves of the subtask in the task breakdown tree. If any capability is required by more than one of the subtask's elemental sub-subtasks, the merit

factors associated with that capability are combined to result in one merit factor for each capability required by the subtask. At the beginning of the problem execution, these merits have initial values. However, as the subtasks are performed, the job planner (not addressed in this paper) can alter the merit factors as necessary after each subtask completion to reflect new knowledge about the tasks. The task allocator would then derive a new allocation based on the adjusted merit factors.

Thus, the task allocator must have information such as that shown in figure 3 concerning the capabilities required to perform a task.

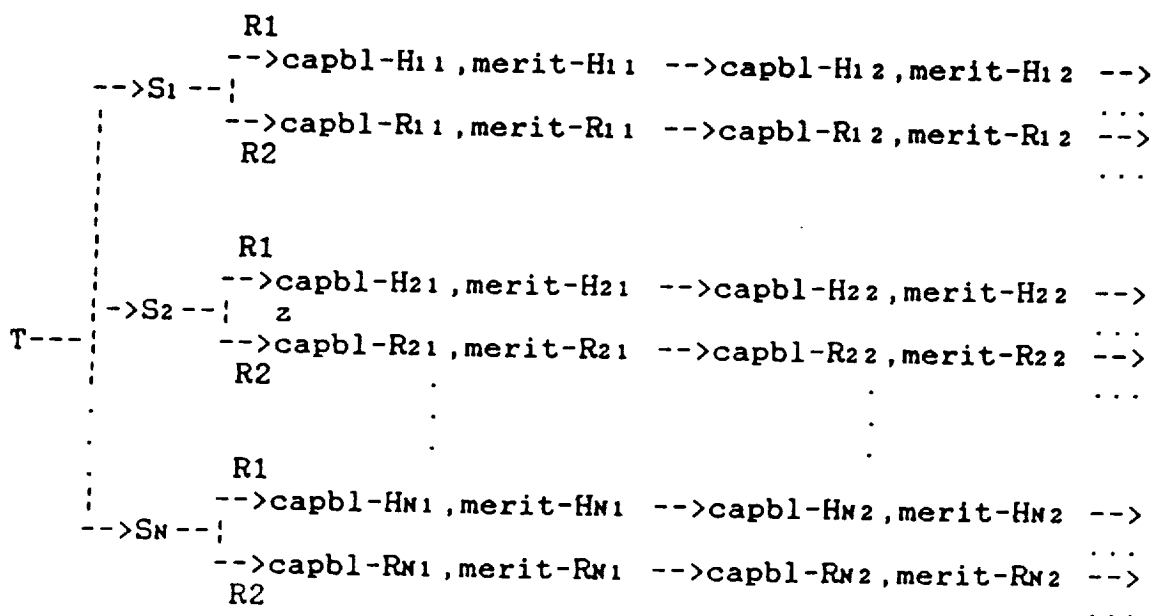


Figure 3

Figure 3 shows that task T consists of N subtasks S1 through SN. For each subtask, the task allocator knows the list of capabilities and merit factors required by each resource to perform the subtask. For example, to perform the subtask S2, the resource R1 must possess capabilities "capbl-H21", "capbl-H22", and so on, which have merit factors of "merit-H21", "merit-H22", and so on. The task allocator can then compare the list of capabilities required for a resource to perform a subtask (the task information) with the actual capabilities possessed by the resource (the resource information) to determine whether the resource is capable of performing the subtask. After completing these comparisons for both resources, the task allocator can obtain the optimal subtask allocation by determining which

resource most suitably meets the constraints/criteria of the problem, and then assigning the subtask accordingly.

Although this paper is addressing the allocation problem requiring only one manipulation subtask to be executed at a time (a sequential-task problem), the extension to several machines and multitasking could be possible with this methodology by incorporating into the task allocator the ability to handle information such as precedence constraints among the subtasks.

3.1.4 ENVIRONMENT

In order to satisfy the constraints and criteria of the problem, the task allocator may often need to have access to information about the environment. The details to be contained in the environmental knowledge base must include information on what is in the environment, what the environment looks like, and how the environment behaves. In addition, the presence of certain environmental conditions may activate certain new constraints/criteria which the task allocator must address.

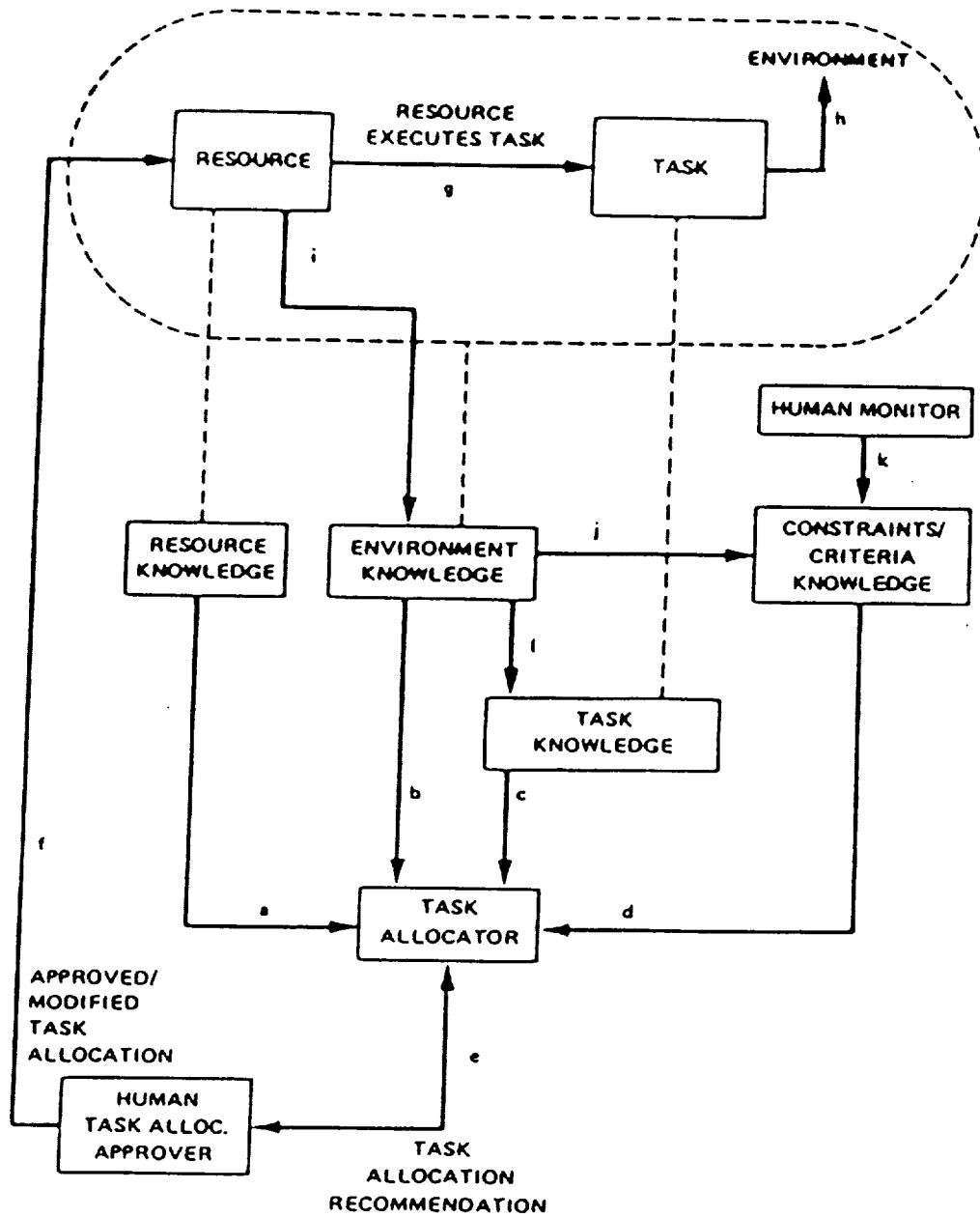
The environmental information will also be accessed by the resources to help them function effectively in their environment. For example, there may be obstacles to avoid or tools available for use in performing a subtask. If the robot were told to GET WRENCH, it must know what a wrench looks like and possibly have an idea of where to find it.

Of course, the human could conclude many things about the environment by simply observing it. However, the computer must operate with an automated representation of its environment. The specific representation of the environment is highly dependent on the application and would thus vary accordingly. Possible representations include frames, rules, scripts, and nets.

3.2 FLOW OF EXECUTION

The current information about the constraints/criteria, resources, tasks, and environment will be stored in separate computerized knowledge bases, and will be shared among all the entities which need the information. These knowledge bases will be kept current by the use of sensors which monitor the resources, the environment, and the tasks, or they could be directly updated by the resources. In order for the man-machine symbiotic system to work effectively, it is important that the knowledge areas be able to interact. Figure 4 depicts the relationship between the knowledge areas.

Fig. 4: PRIMARY INTERACTIONS IN A DYNAMIC TASK ALLOCATION PROBLEM



In figure 4 the dotted oval indicates the actual environment. The three double-dotted lines connecting the resource and the resource knowledge, the environment and the environmental knowledge, and the task and the task knowledge indicate a close association between the physical entities (resource, environment, task) and the knowledge of the entities. The information which can be obtained from either the physical entities or from the knowledge of the entities should be the same.

Figure 4 shows that the task allocator uses knowledge about the resources, environment, tasks, and constraints/criteria (links a, b, c, d) to make a task allocation recommendation. If necessary, the human task allocation approver may change this task allocation (link e). Note that although it is possible that the human task allocation approver is the same person who performs the subtasks, this does not necessarily have to be true. The resource is then assigned a subtask according to the approved/modified allocation (link f). As the resource executes the subtask (link g), the changing subtask status in itself modifies the environment (link h). Possibly, the resource will notice additional events or changes in the environment and will update the environmental knowledge directly (link i). As the environment changes, the constraints/criteria may need to be changed automatically to reflect the new conditions (link j), or manually by a human who monitors the problem execution (link k). Again, the human monitor need not necessarily be the same human who performs the subtasks or who approves the task allocation. Additionally, the list of subtasks to be performed might need to be altered because of environmental modifications (link l). Using the updated knowledge about the resources, the environment, the subtasks, and the constraints/criteria, the task allocator can replan the task allocation as necessary to repeat the cycle.

3.3 DYNAMIC NATURE OF TASK ALLOCATION

One of the key features of this task allocation methodology is its ability to be event-driven, responding to changes in the information about the constraints/criteria, resources, tasks, or the environment by altering the task allocation. Such a dynamic nature of the task allocation is essential to allow the man-machine symbiont to cope with a changing work context. The dynamic nature of the task allocator is directly related to the information in the knowledge bases. If the information in the knowledge bases never changed, the task allocation would never change. However, in a real-world problem, the information in each of

the knowledge bases will be undergoing continual changes to reflect the true state of the problem and the accumulation of experience. The following paragraphs explain how each of the knowledge bases can change.

First of all, although the constraints/criteria are initially set for a particular application, dynamic changes in the work context or environment may cause the constraints/criteria to be changed. The knowledge base changes can be made directly by some type of sensor, or they can be modified manually by a human. For example, the human might decide to change the effective constraint from "minimize time of task completion" to "minimize human involvement" after experiencing fatigue following a long series of manipulation tasks. The task allocator would then allocate the subtasks by attempting to assign as few subtasks as possible to the human.

Secondly, as the resources execute the subtasks, the level of achievement factors and the timeliness-of-achievement factors for their capabilities may change, reflecting new knowledge about the resources. Such changes can take place in two ways: through a learning scheme and through monitoring of the resources. The learning scheme (discussed in a companion paper) allows the robot to learn and improve its capabilities by observing the human. For example, suppose the subtask to be allocated is FIND WRENCH. Initially, the robot will not know what a wrench looks like, indicated by a level of achievement factor of zero or "unknown" for the capability RECOGNIZE WRENCH. The task allocator will therefore assign the subtask to the human, who is then observed by the robot as he performs the subtask. In observing the human, the robot learns what a wrench looks like, and its level of achievement factor is upgraded accordingly. The allocation of the next subtask requiring the ability to recognize a wrench will take into account the new capability factors and will possibly result in a new allocation.

The second method in which the level of achievement factors and the timeliness of achievement factors can change is through monitoring of the resources. It is very important that the knowledge of the resources be consistent with the actual resources themselves. To accomplish this, some type of monitor must observe and quantify the resource's performance to determine if there is a proper correlation between the resource and the knowledge about the resource. If not, the resource knowledge base must be corrected. For example, if the human has a level-of-achievement factor of 7 for the capability MANIPULATION, but does not perform at that level after several hours of work (possibly due to fatigue or boredom), the factor should be

appropriately updated in the knowledge base for use in future subtask allocations.

The information in the third knowledge base, the task information, is subject to change during the execution of the subtasks when environmental changes occur which require the job planner to update the list of subtasks to be performed. The task allocator should recognize these changes and be able to replan the task allocation appropriately. For example, if the event WRENCH DROPPED occurred, the subtask sequence would be reconfigured by the job planner to include the subtask PICK UP WRENCH. The task allocator should then respond to this event and reallocate the subtasks to reflect the change.

Finally, the fourth knowledge base, the environmental information, must be dynamic to allow for changes in the environment, such as successful subtask completion, and for unexpected events, such as subtask failure, to be detected. The changes to the environmental knowledge could come from information supplied directly by the resources, or from sensors separate from the resources. This dynamic feature is important to allow the task allocator to recognize the need for re-allocation of subtasks due to changes in the environment.

4.0 CONCLUSION

A methodological approach for dynamically allocating tasks to a human and an intelligent machine involved in a man-machine symbiotic system has been presented. The necessary knowledge areas and flow of execution have been outlined, and the proposed architecture has been shown to allow dynamic response and task reallocation due to changes in the work constraints, physical environment, and capabilities of the human and the machine, as well as to unanticipated events and human requests or controls. Major man-machine task allocation issues such as event-driven dynamics, knowledge updating through observation and learning, and performance-based work distribution have been discussed. Although this methodology was designed in the context of a remote-manipulation system involving only two symbiotic partners sharing control of a single manipulator arm to accomplish a series of sequential tasks, the methodology has been shown to have the potential for being extended to systems including more than two partners, multitasking operations, or multi-constraint situations. The architecture has been designed to be fully compatible with learning schemes and job-planning methodologies and future work will include the addition of automated monitoring, automated learning, and job planning modules to the current system.

REFERENCES

- [1] Chu, Y., W.B. Rouse, "Adaptive Allocation of Decisionmaking Responsibility Between Human and Computer in Multitask Situations", IEEE Trans. Syst., Man, Cybern., Vol. SMC-9, No. 12, pp. 769-778, 1979.
- [2] Hamel, W.R., Jorgensen, C.C., Weisbin, C.R., "Man-Robot Symbiosis: Schemes for the Evolution of Autonomous Systems", ORNL/TM-10396 (in process).
- [3] Jorgensen, C.C., "Neural Network Recognition of Robot Sensor Graphs using Hypercube Computers", Second Conference on Hypercube Multiprocessors, Sept. 29-Oct. 1, 1986, Knoxville, TN.
- [4] Jorgensen, C.C., "Neural Network Representation of Sensor Graphs for Autonomous Robot Navigation", to be presented IEEE International Conference on Neural Networks, June 21 - 24, 1987, San Diego, CA.
- [5] Jorgensen, C.C., C. Matheus, "Catching Knowledge in Neural Nets", AI Expert, Vol. 1, December 1986.
- [6] Licklider, J.C.R., "Man-Computer Symbiosis", IRE Trans. on Human Factors in Electronics, Vol. HFE-1, pp. 4-11, 1960.
- [7] Parker, L.E., F.G. Pin, "Dynamic Task Allocation for a Man-Machine Symbiosis System", ORNL/TM-10397 (in press).
- [8] Revesman, M.E., J.S. Greenstein, "Application of a Mathematical Model of Human Decisionmaking for Human-Computer Communication", IEEE Trans. Syst., Man, Cybern., Vol. SMC-16, No. 1, pp. 142-147, 1986.
- [9] Rieger, C.A., J.S. Greenstein, "The Allocation of Tasks Between the Human and Computer in Automated Systems", Proceedings of IEEE 1982 International Conference on Cybernetics and Society, pp. 204-208.
- [10] Rouse, W.B., "Human-Computer Interaction in the Control of Dynamic Systems", Computing Surveys, Vol. 13, No. 1, pp. 71-99, 1979.
- [11] Rouse, W.B., "Human-Computer Interaction in Multitask Situations", IEEE Trans. Syst., Man, Cybern., Vol. SMC-7, No. 5, pp. 384-392, 1977.